LECTURE NOTES OF Web Design

By

Dr. Samaher Al_Janabi

Faculty of Science for Women, University of Babylon, Iraq

Samaher@uobabylon.edu.iq

النماذج وكيفية تضمينها في صفحات الويب

<FORM> ..... </FORM>

مع أن النماذج تعتبر من المواضيع المتقدمة (وغير السهلة) نوعاً ما في لغة HTML إلا أن معظم مواقع الويب تكاد لا تخلو من وجودها، وذلك لعدة أسباب لعل من أهمها إيجاد إمكانية للتفاعل بين الموقع وصاحبه من جهة والزوار من جهة أخرى...أحيانا قد تحتاج كمصمم لموقع ويب أن تعرف آراء زوار موقعك في مسائل معينة وقد تكتفي برسائل البريد الإلكتروني التي يرسلوها لك، لكن عندما تريد معرفة أشياء محددة بذاتها فإن النماذج هي الخيار الأفضل لك. بالإضافة إلى إمكانية تنظيم البيانات المدخلة من خلالها وسهولة وسرعة استخدامها من قبل زوار الموقع. ومن أبرز الأمثلة على النماذج في مواقع الويب هي دفاتر الزوار وصفحات البحث عن الكلمات أو العبارات داخل المواقع.

وهي مجرد وسوم عادية مثلها مثل HTMLلا تكمن صعوبة التعامل مع النماذج في كونها معقدة بحد ذاتها، كلا ... فهي إحدى العناصر التي تدعمها لغة الوسوم التي تعاملنا معها في جميع الدروس السابقة. وبإمكانك إنشاء النماذج في موقعك بنفس السهولة التي تدرج فيها جدولاً أو إطاراً (هذا بالطبع إذا كنت تعتقد أن الجداول والإطارات سهلة) لكن التداخل بينها (وأعني النماذج) وبين لغات البرمجة المتقدمة في الويب مثل JavaScript, CGI هي ما يجعلها تختلف عن سابقيها من الوسوم أو العناصر الأخرى. خاصة إذا احتجت إلى بعض المقاطع البرمجية من هذه اللغات ضمن نماذجك. أما إذا اكتفيت بالنسبة للنماذج. فما من مشكلة... لأنه سيكون بإمكانك التعامل معها بكل بساطة. HTMLبالإمكانات المتواضعة التي توفرها

كم شكلاً من أشكال إدخال البيانات؟

الجواب هنالك العديد منها وليك ست انواع منها تتوفر غالبا في دفتر الزوار


Text
أر سل
اذ سى الأمر

يجب أن تدرج جميعاً ضمن وسمين أساسيين للنماذج هما:

<FORM> ... </FORM>

وكما جرت العادة نحتاج لتحديد بعض الخصائص التي تتعلق بطبيعة هذا النموذج. ولدينا هنا ثلاث خصائص:

1. ACTION

تحدد العنوان الذي سيتم إرسال بيانات النموذج إليه لتتم معالجتها بالصورة المطلوبة. وعادة يكون هذا عنواناً لبريد إلكتروني Email سوف يتم إرسال بيانات النموذج إليه. أو قد يكون عنواناً لبرنامج CGI موجود على الكمبيوتر الخادم Serverالذي تتواجد عليه صفحة الويب، حيث يستقبل هذه البيانات ويعالجها حسب التعليمات الموجودة فيه كأن يضيفها مثلاً إلى إحدى الصفحات (كما يحدث عادة في دفاتر الزوار) أو يتحقق من صحة بعض الحقول المدخلة ومطابقتها لمعايير معينة، أو أن يقوم بالبحث عن كلمة أو عبارة ضمن صفحات الموقع كما في نماذج البحث الموجودة في مواقع الويب.

```
<FORM ACTION="mailto:someone@domain.com"> ... </FORM>
<FORM ACTION="name_and_address_of_CGI_script"> ... </FORM>
```

2. METHOD

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة ACTION. وهناك قيمتين لهذه الخاصية هما: GET التي تستخدم في حالة كون عملية المعالجة داخلية أي تتم داخل الخادم Server نفسه. ففي مثالنا السابق عندما نستخدم نموذج البحث عن كلمة في الموقع، فإن عملية المعالجة (أي البحث) تجري مباشرة في الموقع. والقيمة الثانية هي Post وتستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد إلكتروني.

```
<FORM ACTION="mailto:someone@domain.com" METHOD="post"> ... </FORM>
<FORM ACTION="name_and_address_of_CGI_script" METHOD="get"> ... </FORM>
```

3. ENCTYPE

هذه الخاصية تحدد طريقة الترميز التي سيتم إرسال البيانات وفقاً لها. وهي تأخذ القيمتين التاليتين: (يجب أن تكتب هذه القيم كما هي نصاً وحرفاً)

- application/x-www-form-urlencoded
- text/plain

وبدون الخوض في الأسباب التقنية التي أدت إلى إيجاد هذين النوعين من طرق الترميز أو في أمور برمجية بعيدة عن موضوعنا، فإن الدافع لإستخدام أي من القيمتين هو طبيعة عملية المعالجة التي ستجرى على البيانات أو طبيعة برنامج البريد الإلكتروني الذي ستستقبل هذه البيانات من خلاله (إذا كان يدعم MIME أم لا، وهي إختصار للعبارة Multi-purpose Internet Mail Extentions وهي من المعايير السائدة في الإنترنت والتي تتعلق بنقل جميع أنواع البيانات من صوت وصورة وليس فقط النصوص من خلال البريد الإلكتروني). وما يعنينا هنا هو الفرق بين الطريقتين من حيث طريقة إرسال واستقبال البيانات. فعند استخدام text/plain ستصل البيانات بالشكل التالي:

```
NAME=Yahya Al-Sharif
Address=Nablus , Palestine
Email=yahya@palnet.com
```

(الكلمات Name, Address, Email هي أسماء الحقول في النموذج ونقوم نحن بتعريفها أثناء عملية تصميم النموذج أما النصوص الظاهرة بعد إشارة المساواة فهي البيانات المدخلة، وسوف نتحدث عن تعريف أسماء الحقول بعد قليل)

أما عند استخدام **application/x-www-form-urlencoded** فستصل البيانات بالشكل:

```
NAME=Yahya+Al-Sharif&Address=Nablus+,+Palestine&Email=yahya@palnet.com
```

ولك أن تخيل مبلغ الصعوبة في تحليلها إذا احتوت على عشرات الحقول. لذلك تتوفر برامج خاصة تعرف بـ Formaters تقوم بإعادة ترتيب البيانات المرسلة من خلال النماذج بشكل مفهوم بحيث تصبح كما لو كانت مرسلة بترميز text/plain وإليك أحدها وهو برنامج مجاني يدعى UrlCook. لكن لا تعتقد أن الطريقة الأولى هي الأفضل دائماً فذلك يعتمد كما قلنا على طريقة المعالجة والنقل بالبريد. لذلك لا ضير من أن تجرب الطريقتين لتعرف أيهما أنسب لك.

إذن خلاصة القول: قد تكون أفضل صيغة لتعريف النموذج في حالة أردت استقبال البيانات من موقعك إلى عنوان بريدك الإلكتروني هي:

```
<FORM ACTION="mailto:email@domain.com" METHOD="post" ENCTYPE="text/plain">
...
</FORM>
```

وبهذا نكون قد إنتهينا من عملية تعريف النموذج وخصائصه، لكن انتظر فما زلنا في بداية الطريق.

نبدأ الآن في عملية تعريف أشكال البيانات في النموذج. ونستخدم الوسم **<INPUT>** لتعريفها والحقيقة أن هذه الأشكال هي مجرد خصائص أو بالأحرى قيم لخصائص تابعة لهذا الوسم. كيف؟ ... لنأخذ مثالاً على ذلك لأوضح لك هذا المفهوم

```
<FORM ...>
<INPUT TYPE="text">
</FORM>
```

لينتج لدينا هذا الشكل:

فقرة معترضة:

إليك جميع الأشكال (القيم) المستخدمة مع الخاصية TYPE وسوف أتركها الآن بدون تعليق لحين مناقشها لاحقاً بشكل مفصل. مع ملاحظة أن هناك شكلين آخرين ندرجهما بالوسوم <SELECT>, <TEXTAREA>

<INPUT TYPE="text">	
<INPUT TYPE="password">	**********
<INPUT TYPE="hidden">	
<INPUT TYPE="radio">	
<INPUT TYPE="checkbox">	
<INPUT TYPE="submit">	Submit
<INPUT TYPE="reset">	Reset
<INPUT TYPE="button">	

أرجو أن أكون قد وضحت لك الآن وظيفة الخاصية TYPE وجميع القيم المستخدمة معها
ونعود الآن إلى مثالنا.. الخاصية الثانية المستخدمة مع <INPUT> هي NAME وتستخدم لتسمية حقل البيانات حيث قمت بإعطاء الإسم address لهذا الحقل في المثال. (لك كل الحرية في إعطاء الإسم الذي تريده للحقل). والحقيقة أن هذا الإسم يعرّف الحقل في داخل النموذج نفسه، بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي قد تضيفها كمصمم للموقع. وحتى عندما تريد أن يُرسل النموذج إليك بالبريد فإن حقوله تعرّف بالاسم الذي أدرجته لها من خلال هذه الخاصية. (لاحظ ما قلتُه سابقاً عن تعريف أسماء الحقول عندما تحدثنا عن الترميز والطرق التي تصل بها محتويات النموذج). وكما ترى لا يوجد (حتى الآن) ما يدل على أن هذا الحقل يختص بإدخال العنوان.

```
<FORM ...>
<INPUT TYPE="text" NAME="address">
</FORM>
```

خصائص الوسم INPUT هي:

- **TYPE**: لتحديد نوع (شكل) حقل البيانات.
- **NAME**: لتعيين اسم لحقل البيانات.
- **VALUE**: لتعيين قيمة إفتراضية (مبدئية) لحقل البيانات.
- **SIZE**: لتحديد حجم حقل البيانات.
- **MAXLENGTH**: لتعيين الحد الأقصى لعدد الحروف المدخلة في الحقل.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text" NAME="address" VALUE="Nablus, Palestine"
SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your address : Nablus Palestine

من حيث الخصائص تماماً غير أن مدخلاته تظهر على text وهو يشبه الحقل password النوع الثاني من الحقول المستخدمة في النماذج هو حقل مهما كانت، وهو الفرق الوحيد بينهما. وربما تكون قد استنتجت الآن أن هذا النوع من الحقول يستخدم عندما يوجد حاجة لإدخال كلمة سر من قبل ****** شكل الزائر في النموذج

```
<FORM ...>
    Please enter your name :
        <INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
    Please enter your passwod :
        <INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your name :

Please enter your password : ******

للحقول، ولذلك تركتها فارغة وأستطيع أيضاً أن ألغيها نهائياً من الشيفرة. وأنا في هذا المثال أردت أن VALUES لاحظ أنني لم أرغب في كتابة قيم إفتراضية أوضح لك عدم أهمية كتابة قيمة إفتراضية للحقول في بعض الحالات.

نأتي الآن إلى النوع الثالث من أنواع الحقول وهو hidden أي الحقل المخفي. وكما نستنتج من اسمه فهو لن يظهر ضمن النموذج. وهذا مثال:

```
<FORM ...>
        Please enter your name :
                <INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
                 <INPUT TYPE="hidden" NAME="my forms" VALUE="form>

        Please enter your passwod :
                <INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your name :
Please enter your passwod :

لاحظ هنا أن وجود هذا الحقل مثل عدمه بالنسبة لمظهر النموذج، وأن الزائر لن يتعامل معه بل وربما لن يعرف أن هناك حقلاً مخفياً. والسؤال هنا: ما الفائدة من وجود شيء مخفى لا إمكانية لاستخدامه؟ ولكي أجيب على هذا السؤال دعني أطرح لك مثالاً أو حالة قد تواجهك كمصمم صفحات ويب...
لنفرض أن لديك ثلاث صفحات تتضمن كل منها نموذجاً ما وأن هذه النماذج متشابهة. وتحتوي على نفس الحقول. وعندما تصلك البيانات كيف ستستطيع تمييز أي من هذه النماذج استخدم لإرسال البيانات؟ بإمكانك إضافة هذا الحقل (الوهمي) وإسناد أي اسم وأي قيمة له في كل نموذج.

في النموذج الأول ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">
```

في النموذج الثاني ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE=" form2">
```

في النموذج الثالث ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE=" form3">
```

وبذلك عندما تصلك البيانات المرسلة من قبل أي زائر استخدم أي من النماذج الثلاثة سيصلك أيضا حقل إضافي قمت أنت نفسك بتعبئته سلفاً عندما صممت النموذج وذلك بأحد الأشكال التالية:

my forms=form1 أو
my forms=form2 أو
my forms=form3

ملاحظة مهمة بالنسبة للنماذج بشكل عام. من أجل إظهار النموذج بصورة مرتبة ومنسقة والتحكم بموقع الحقول فيه فمن الأفضل دائماً وضعه داخل جدول مع جعل الجدول بلا حدود.

```
<FORM ...>
    <TABLE BORDER="0">
        <TR>
            <TD>Please enter your name : </TD>
            <TD>
                <INPUT TYPE="text" NAME="the name" VALUE="" SIZE="40" MAXLENGTH="30">
            </TD>
        </TR>

        <TR>
            <TD>Please enter your password :</TD>
            <TD>
                <INPUT TYPE="password" NAME="the password" VALUE="" SIZE="40" MAXLENGTH="30">
            </TD>
        </TR>
    </TABLE>
</FORM>
```

وكما ترى تحتاج إلى القليل من العمل الإضافي لكنك بالمقابل ستحصل على النتيجة التالية

Please enter your name :	
Please enter your password :	